Cost-Benefits of Automation for Surface Operations: Preliminary Results

Efforts to automate missions involving unmanned spacecraft have resulted in significant quantified benefits. Ground systems automation of the Mars Exploration Rover (MER) led to a 15 - 40% increase in efficiency. Ground systems automation of the Space Technology-5 mission and onboard automation for the Earth Observing-1 mission each led to cost savings in excess of 20%. In addition to these operational successes of automation, numerous technology demonstrations have shown additional potential benefits of software and automation. Experimental adaptive control technology flown on F-15 Active, F-18 and simulated for use on a C-17 uses redundant control to help pilots continue to fly damaged aircraft, with potential increases in safety.

Despite these successes, software is still considered a risk, particularly in human spaceflight. The 1996 Arianne V launch failure was ultimately traced to software problems, one of which led to an erroneous command to the engines, the second of which caused both Inertial Reference Systems to fail. The Mars Climate Orbiter (MCO) was lost in 1999 due to inconsistent engineering unit representations in mission operations that were not detected until late in the flight. The DART mission in 2005 failed due to a combination of GNC software flaws. Software faults are not limited to space flight; in 2007 a software flaw caused a failure of the F-22 navigation software during the aircraft's sortie beyond the International Dateline. The risks of software for NASA are considered unacceptable given the potential for human astronaut casualties; when considered in conjunction with growing software costs, there has been a tendency to avoid heavy use of automation software in current NASA projects.

As NASA's Project Constellation develops requirements and operations concepts to enable the next human visits to the moon, the Constellation Surface Systems office undertook a series of software cost-benefits studies. The purpose of these studies was to quantify the costs and benefits of software with regard to the design and operation of the Lunar Lander (Altair), the Lunar habitat, and other surface assets.

The Automation For Operations (A4O) project, funded by NASA's Exploration Technology Development Program, has prototyped an evolvable mission operations architecture supporting the operations of manned space vehicles, unmanned space vehicles, surface assets and robotic systems. The goals of the project are to achieve quantitative reductions in operations costs and increases in operations efficiency, while maintaining safety. The A4O project studied the costs and benefits of automating three capabilities for which software was identified as having a significant up-front and recurring cost, but could also provide significant benefits; Extra-Vehicular Activity (EVA) support, Quiescent Operations, and Cargo Offload. The project staff analyzed current mission operations tools for each of these three capabilities and emerging technologies and classified each technology as enabling a "low, medium" or "high" degree of operations automation. Technology costs were estimated using a traditional line-of-code count. Technology benefits for each technology were identified as: person hours required to build plans, staff to build plans, turnaround for modifying plans, unproductive crew hours, and staff in mission control to monitor subsystems. These benefits were aggregated into more abstract metrics relevant for surface operations: cost, mass, safety.

The study shows that automation can have significant benefits for cost and safety. Cost decreases are derived from reductions in staff hours to build plans and monitor subsystems, reductions in process time, and unproductive crew hours. Safety increases due to enhanced monitoring ultimately lead to cost reductions, since safety will never be compromised by staff reductions. While mass reductions are difficult to quantify, the use of automation is argued to lead to reductions in mass for certain restricted scenarios.

The study also shed light on the complexity of estimating software cost. The study reveals that autocoding techniques generate significant numbers of lines of code, which may skew traditional cost estimates. The study reveals that modern software technology can lead to significant cost savings when up-front commitments to software interoperability and reconfigurability are made. However, if software configuration is more than just parameters, such configurations may be better counted as lines of code.